

Research Article

Construction of a Smart Medication Dispenser with High Degree of Scalability and Remote Manageability

JuGeon Pak and KeeHyun Park

Computer Engineering Department, Keimyung University, Sindang-dong, Dalseo-gu, Daegu 704-701, Republic of Korea

Correspondence should be addressed to KeeHyun Park, khp@kmu.ac.kr

Received 24 May 2012; Revised 2 June 2012; Accepted 2 June 2012

Academic Editor: Sabah Mohammed

Copyright © 2012 J. Pak and K. Park. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We propose a smart medication dispenser having a high degree of scalability and remote manageability. We construct the dispenser to have extensible hardware architecture for achieving scalability, and we install an agent program in it for achieving remote manageability. The dispenser operates as follows: when the real-time clock reaches the predetermined medication time and the user presses the dispense button at that time, the predetermined medication is dispensed from the medication dispensing tray (MDT). In the proposed dispenser, the medication for each patient is stored in an MDT. One smart medication dispenser contains mainly one MDT; however, the dispenser can be extended to include more MDTs in order to support multiple users using one dispenser. For remote management, the proposed dispenser transmits the medication status and the system configurations to the monitoring server. In the case of a specific event such as a shortage of medication, memory overload, software error, or non-adherence, the event is transmitted immediately. All these operations are performed automatically without the intervention of patients, through the agent program installed in the dispenser. Results of implementation and verification show that the proposed dispenser operates normally and performs the management operations from the medication monitoring server suitably.

1. Introduction

Medication nonadherence is a serious public health issue with the increase in chronic diseases [1–3]. To improve patient adherence, medication dispensers are often proposed [4–8]. A medication dispenser is a device that delivers medication to the patient according to predetermined schedules; it is considered a very efficient device of improving medication adherence [9–12].

However, there is some room for improvement in the existing medication dispensers. (1) Most existing medication dispensers are designed to support only a single user and have a low degree of scalability. Thus, the assignment of one medication dispenser to each patient would increase the operation costs. (2) A correct medication schedule and system settings should be preconfigured in a medication dispenser. However, most existing medication dispensers require users to configure the schedule and settings manually. This leads to inconvenience and errors due to mistyping. (3) An error in a medication dispenser can have fatal

consequences. Nonetheless, existing medication dispensers are not equipped with remote device management functions. Therefore, users must manage their medication dispenser by themselves.

Moreover, considering that users of medication dispensers are typically in the older age group or are mostly patients with chronic diseases, important investigations should be carried out for achieving the improvements mentioned above.

In this paper, a smart medication dispenser with a high degree of scalability and remote manageability is proposed and constructed. The proposed smart medication dispenser allows multiple users to use a single medication dispenser, thus providing scalability to the device. It also allows medical staff and system administrators, instead of end users, to manage medication dispensers, thus leading to cost efficiency and safe operation of the device. Medications for each patient are stored in a medication cartridge and a cartridge is placed in a Medication dispenser tray (MDT). One smart medication dispenser has basically one MDT, but it can be

extended depending on the number of patient (six MDTs maximum). The medication schedule configured in the dispenser is updated remotely by medical staff workers. Also, the system settings, embedded programs, and operational errors are managed remotely by system administrators.

Meanwhile, the smart medication dispenser corrects a patient's medication state and transmits the corrected data to the medication-monitoring server. When an abnormal state is detected by the smart medication dispenser, the dispenser and the server exchange several management messages. Consequently, it can be inferred that the smart medication dispenser requires more frequent message transmission than existing medication dispensers. This causes serious constraint for medication dispensers which operate on limited bandwidth networks. To overcome this constraint, the Open Mobile Alliance (OMA) Device Management (DM) protocol [13] is applied. This protocol is considered a de facto international standard for mobile device management. OMA DM can provide an appropriate solution to the management of the medication dispenser because it was originally designed to accommodate limited bandwidth networks.

The contribution of this paper is the presentation of a smart medication dispenser. The proposed dispenser has three advantages over existing medication dispensers. (1) To achieve a high degree of scalability, the medication-dispensing trays can be attached in succession, and therefore, a single dispenser can support multiple users. (2) To achieve a high degree of remote manageability and (3) to reduce management costs and efforts, remote management methods are designed and implemented. These methods facilitate updating of the medication schedule configured in the smart dispenser. Further, system settings, embedded programs, and operational errors can be remotely managed by medical staff and system administrators.

2. Related Works

To improve medication adherence, various methods based on information technology are being carried out. The method we have found in our survey can be categorized into three types: application-level medication reminders, sensor-based intake trackers, or electronic medication dispensers.

Examples of the application-level medication-reminding method are UbiMeds [14], Wedjat [15], and MyMediHealth [16]. These applications run on mobile devices such as PDAs and smart phones. They provide user interfaces for configuring medication schedules and alert users to the time and type of medication according to the configured medication schedule. They can prevent underdosing and misdosing and are relatively low cost.

Sensor-based intake-tracking methods track and assess the user's medication habits using sensors. SmartDrawer [17], which is a representative example of this type of method, involves use of an RFID tag and reader. Additionally, motion detection technologies based on computer vision are used in some systems [18, 19]. These methods have an advantage in that they can detect whether the user is actually taking the medications.

Finally, electronic medication dispensers are considered very efficient for improving medication adherence [9–12]. They prevent overdosing, misdosing, and underdosing through lockdown of the medication-dispensing tray, dispensing of medications according to the preconfigured medication schedule, and the medication time alarm. Although early medication dispensers were built as stand-alone models that could not communicate with external devices [4, 5], communicable medication dispensers have been proposed in recent years [6–8]. These dispensers collect a patient's medication status and transmit them to a monitoring server to be analyzed by medical staff.

The medication monitoring system described in this paper is similar to these dispensers in terms of transmitting medication status to a remote monitoring server. However, the distinguishing feature of the medication monitoring system proposed in this paper is that it can manage medication dispensers remotely. The medication schedule configured in the dispenser is updated remotely, and the system settings, software, and errors are managed remotely instead of patients.

The Simple Network Management Protocol (SNMP) of the Internet Engineering Task Force (IETF) [20], Web-Based Enterprise Management standard (WBEM) [21] of the Distributed Management Task Force (DMTF), and OMA DM [13] are the representative remote device-management methods that increase device reliability and minimize user inconvenience. Among these methods, OMA DM is the international de facto standard for mobile device management and is used most widely. Various studies are being carried out to apply OMA DM to a wide variety of fields. In the early days, many studies focused on mobile device management [22–24], whereas recent work has focused on software fault management and debugging [25, 26], network management [27], vehicle management [28], and so on. Nevertheless, the application of OMA DM to managing personal health devices is rare.

Considering that the medication dispenser users are typically in the older age group, it is difficult for them to manage their dispensers and to configure numerous settings. Moreover, a medication dispenser closely correlates with the user's health, and an error in a medication dispenser can have fatal consequences. Therefore, extensive research on the reliable maintenance of medication dispensers is essential.

3. Smart Medication Dispenser

The smart medication dispenser proposed in this paper is a component of a medication monitoring system. The medication monitoring system is comprised of smart medication dispensers and a medication monitoring server. Figure 1 depicts the overview of the medication monitoring system.

The smart medication dispenser transmits three types of data to the medication monitoring server: the patient's medication state, device state, and the system settings. The server analyzes the medication state and generates management operations for updating the medication schedule if necessary. The server also analyzes device state and system settings

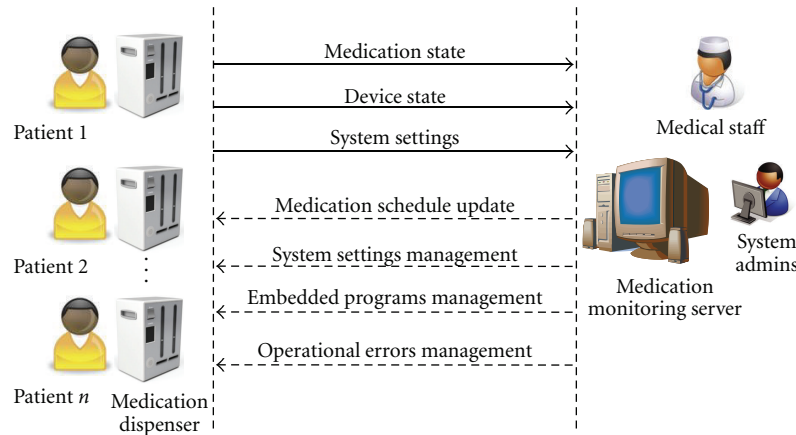


FIGURE 1: Overview of the medication monitoring system.

and generates operations for managing system settings, embedded programs, and operational errors.

3.1. Hardware Architecture. We develop the smart medication dispenser on the WinCE platform. Figure 2 shows the hardware architecture of the dispenser.

- (i) *MCU*: the MCU controls all system functions.
- (ii) *Touch-sensitive LCD*: the liquid crystal display (LCD) displays the medication information. It is also used (by the user) to set parameters such as the medication schedule.
- (iii) *Medication dispensing tray (MDT)*: the MDT contains the medication to be taken. It dispenses the medications when the user presses the Dispense Button at the proper time. The number of MDTs can be extended to six to support multiple users.
- (iv) *Alarm module*: the alarm module notifies the user by a buzzer that it is time to take medications. Upon pressing the Dispense Button, the buzzer terminates and the LED stops blinking.
- (v) *Dispense button*: the dispense button is used to dispense the user's medication. It only works once during each predetermined cycle period.
- (vi) *Infrared sensor*: the infrared sensor checks the quantity of the remaining medications in the MDT. If the quantity drops below the minimum level, it is reported to the medication monitoring server.
- (vii) *Real-time clock*: the real-time clock is used to ensure that the device alerts the user to take his or her medication at the proper time. It is basis for the alarm cycle.
- (viii) *Communication module*: the communication module is used to communicate with a MS or PC. RS232 Serial communication and a local area network (LAN) are provided.

The medication dispenser operates as follows: (1) When the real-time clock reaches the predetermined medication

time, and then (2) the user presses the dispense button at that time, (3) the predetermined medications are dispensed from the medication dispensing tray (MDT).

3.2. Software Architecture. The smart medication dispenser transmits the medication status periodically. If a specific event such as a shortage of medication, medication jam, memory overload, software error, or nonadherence occurs, the event is transmitted immediately. All these operations are conducted automatically without the intervention of patients through software installed in the smart medication dispenser. The software architecture is described in Figure 3.

3.2.1. DM Agent. The DM Agent, which is placed in the smart medication dispenser, manages the dispenser according to the operations of the medication monitoring server. It consists of a Session Manager, Authentication Manager, Protocol Manager, DM Function Manager, and Tree Manager.

The Session Manager manages a management session with the monitoring server. It maintains connections with the server until a session completed. The Authentication Manager generates a user's authentication information to be used when connecting to the server and confirms the identity of a server before performing management operations. The Protocol Manager generates and analyzes the exchanged messages according to specific message encoding methods (XML, WBXML). It constructs a message from the patient's medication status and device status/settings to extract management operations from a received message. The DM Function Manager practically manages the smart medication dispenser in accord with the monitoring server's management operations. It provides the five functions of medication state transmission, medication scheduling, system settings, embedded programs management, and operational errors management. The manageable data in the smart medication dispenser are called management objects (MOs), which are constructed as a so-called DM Tree. The Tree Manager extracts and modifies the values of MOs according to the requests of the DM Function Manager.

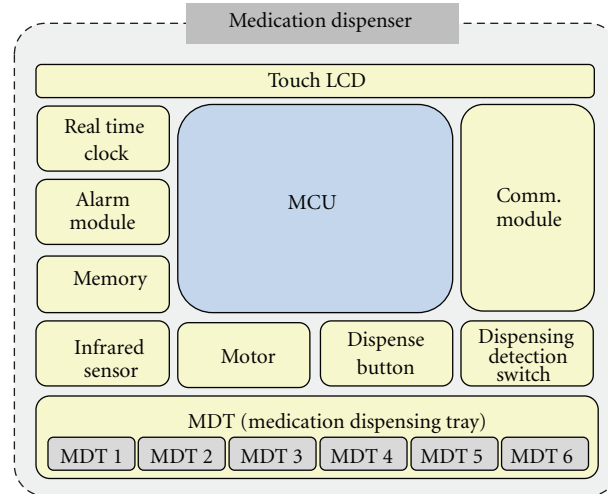


FIGURE 2: Hardware architecture of the smart medication dispenser.

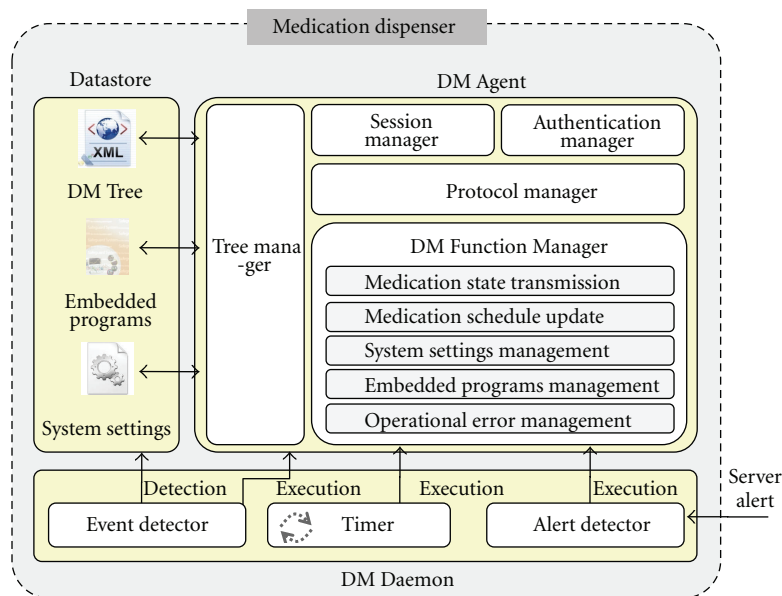


FIGURE 3: Software architecture of the smart medication dispenser.

3.2.2. Datastore. Datastore is a physical memory installed in the smart medication dispenser. The DM Tree (Tree file), embedded programs, and system settings (Configuration file) are stored in the Datastore. The MOs of the DM Tree are transmitted and modified by the DM Agent, and the system settings are updated manually by the user. The DM Daemon synchronizes two files whenever one is changed.

3.2.3. DM Daemon. If the DM Agent were to run constantly, it would be very inefficient in terms of power consumption and resource use. Therefore, we designed the DM Agent in a way that it remains in the sleep mode until a specific event occurs. If an event occurs, then the DM Daemon runs the DM Agent to establish a management session with the medication monitoring server. The DM Daemon is a program that runs in the background. Its primary aims are

to (1) detect specific events, (2) count transmission intervals of medication status, (3) listen to the server's management operations, and (4) execute the DM Agent to establish a management session with the monitoring server. It consists of an Event Detector, Timer, and Alert Detector.

Once the user modifies the system settings manually or an operational error occurs in the smart medication dispenser, the event is registered in the configuration file. While continually detecting the occurrence of specific events, the Event Detector executes the DM Agent if an event is detected. The Timer runs the DM Agent when it reaches the specified interval from the most recent medication status transmission. In general, a management session is initiated at regular intervals by the DM Agent. The monitoring server can also request the DM Agent to initiate a management session by sending a particular message at a specific time.

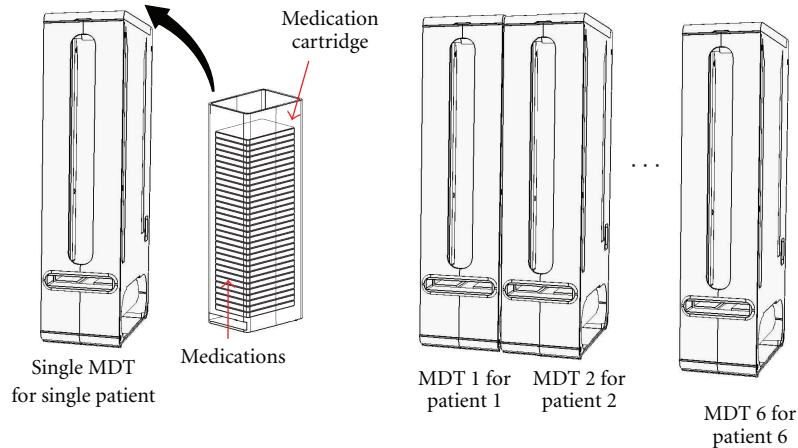


FIGURE 4: Method for increasing the number of MDTs in the smart medication dispenser.

This message is called a Server Alert Message. The Alert detector listens for server alert messages and executes the DM Agent if a message is received.

4. Characteristics of Smart Medication Dispenser

4.1. Multiuser Environment for a High Degree of Scalability. At a hospital or a nursing home, a number of patients might require medication treatments. The assignment of one medication dispenser to each patient would increase the operation costs. In contrast, the proposed smart medication dispenser supports a multiuser environment. Medications for each patient are stored in a medication cartridge, which in turn is placed in an MDT. One smart medication dispenser typically has one MDT, but the dispenser can be extended to include more MDTs, depending on the number of patients (a maximum of six MDTs per dispenser). Figure 4 shows the method for increasing the number of MDTs in the smart medication dispenser.

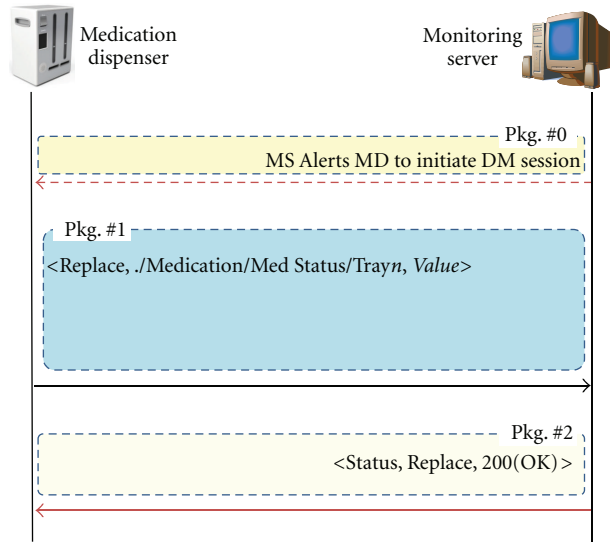
4.2. Remote Management Operations for a High Degree of Manageability. The smart medication dispenser proposed in this paper provides remote management methods. In addition, the management methods are compatible with the OMA DM protocol. Thus, numerous OMA DM servers, which are already widely used in device management, can be utilized as a medication monitoring server. To achieve this, we defined the manageable data of the smart medication dispenser as MOs and arranged them into a tree structure called a DM Tree. The medication monitoring server manages the smart medication dispenser through management operations that contain several commands such as ADD, DELETE, REPLACE, and GET. That is, the medication monitoring server modifies specific MOs using management commands. For example, if a management operation that contains the command “REPLACE” with a specific value targets the MO “Rep.Interval,” this reflects an attempt by the medical staff to change the transmission interval to a desired

value. This is an example of a configuration management operation. We have designed the following management operations: medication status transmission; configuration management; software management; error reporting. For these operations, the smart medication dispensers and the medication monitoring server exchange several messages during a management session.

A management session starts at regular intervals (the value of the MO./Medication/System_Conf/Rep.Interval), whenever an error occurs (whenever the MO./Medication/Error is generated), or when the Monitoring Server alerts the medication dispenser to initiate a management session with Pkg. #0 (Server alert message). When a management session begins, the child MOs belonging to the MO./Medication/Med_Status are transmitted to the Monitoring Server automatically, whereas the child MOs belonging to the MOs./Medication/User./Medication/Med_Schedule./Medication/Sys_Conf, and ./Medication/SW are transmitted only when the medication monitoring server requests them. On the other hand, if an error occurs, then the dynamic MO./Medication/Error is generated and transmitted immediately. Figure 5 depicts the management operations for the smart medication dispenser.

4.2.1. Medication State Transmission. Figure 5(a) shows the medication state transmission operation. The medication state is stored in the MO./Medication/Med_Status/Tray as six characters. That is, the smart medication dispenser can distinguish up to six different medications. Each character represents the medication state (0: nonadherence, 1: adherence, 2: forcible dispensing, 9: upcoming). For example, the MO./Medication/Med_Status/Tray1 with the value “012999” represents the following facts: (1) the patient used the first medication tray; (2) s/he did not take the first type of medication; (3) s/he took the second type of medication; (4) s/he dispensed the third type of medication forcibly; (5) s/he is expecting to take fourth, fifth, and sixth types of medications later.

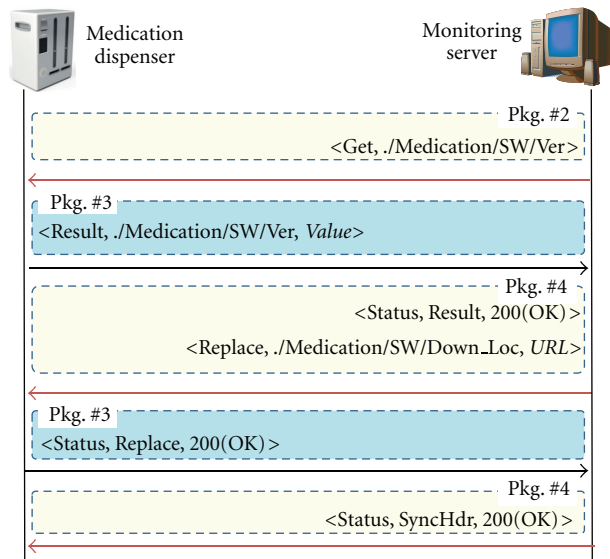
As described in Figure 5(a), the smart medication dispenser transmits the medication state with pkg. #1, which



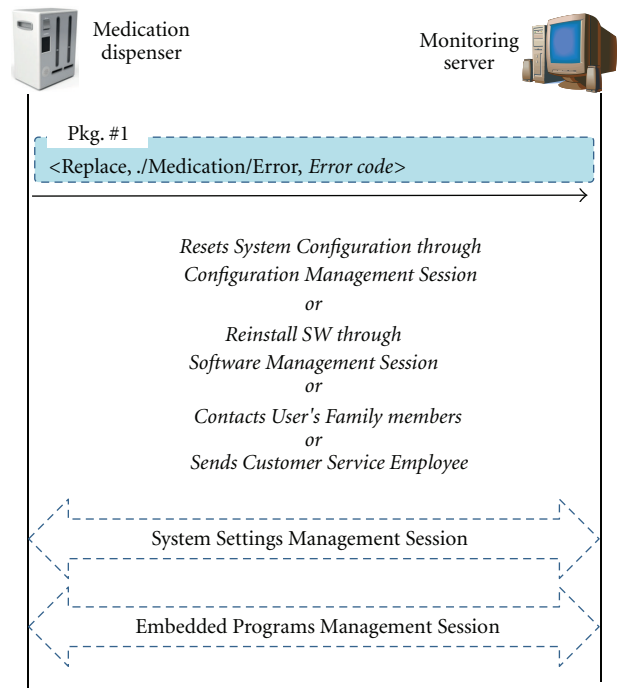
(a) Medication state management



(b) System settings management



(c) Embedded programs management



(d) Operational errors management

FIGURE 5: Management operations.

contains the REPLACE command when the server requests it or the timer reaches the transmission time. The medication monitoring server stores the received medication state in its

Datastore, and transmits pkg. #2 including the status code 200 (OK) as a response. During these processes, the stored medication state is provided to the medical staff.

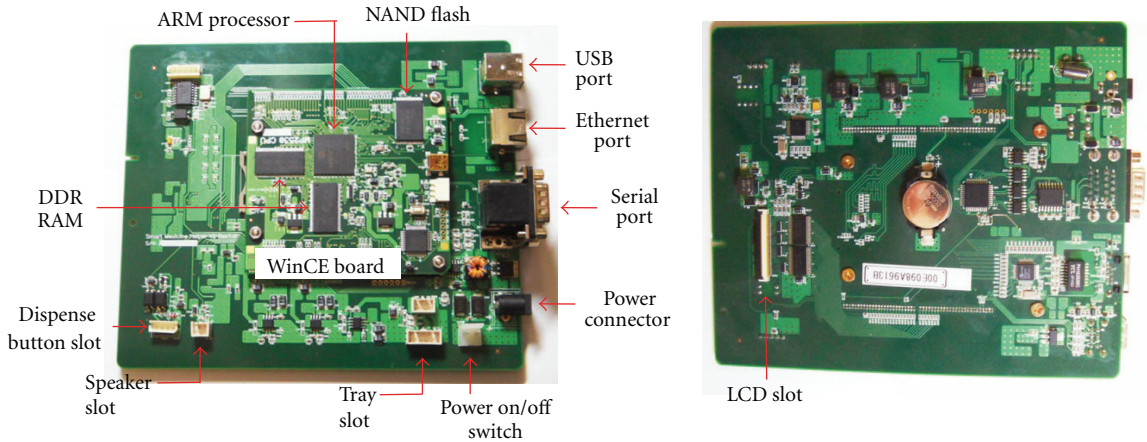


FIGURE 6: PCB of smart medication dispenser.

4.2.2. System Settings Management. Figure 5(b) shows the system settings management operation. The medication monitoring server alerts the smart medication dispenser when it seeks to manage specific system settings. Before managing the settings, the Monitoring Server first checks the assigned values of the target settings using GET commands. The medication dispenser extracts the assigned values that are requested and returns them with RESULT commands. The medication monitoring server analyzes the received values and then sends management commands to add, delete, or replace the target settings. As shown in Figure 5(b), the medication monitoring server sends the new medication schedule and transmission interval using REPLACE commands. The medication dispenser modifies the existing schedule and interval to the received values and, thereafter, operates according to the updated schedule and interval.

4.2.3. Embedded Programs Management. Figure 5(c) shows the embedded programs management operation. When a new version of embedded program is released, the medication monitoring server makes the smart medication dispenser update its program. To accomplish this, the server alerts the smart medication dispenser to initiate a management session and requests information on the version of installed program using a GET command. The medication monitoring server then analyzes the version information returned by the smart medication dispenser. If the version is found to be old, the medication monitoring server starts updating the program of the medication dispenser. It sends the URL for updating the program along with a REPLACE command to the medication dispenser. The medication dispenser then connects to the URL and downloads the new program automatically. After the completion of downloading, the medication dispenser installs the program automatically and returns the result of the update.

4.2.4. Operational Errors Management. Figure 5(d) shows the operational errors management operation. If an error occurs in the smart medication dispenser, a dynamic MO./Medication/Error is generated with a specific error

code. We defined four types of error codes (1: shortage of medication, 2: medication jam, 3: memory overload, and 4: software error). Once the DM Daemon running in the medication dispenser detects an error, it runs the DM Agent, which transmits the corresponding error code to the Monitoring Server by using a REPLACE command. The Monitoring Server analyzes the error code, and informs the patient and guardians of the fact in the case of a medication shortage. In the case of a medication jam, the guardians are notified of the error; in the event of memory overload, the Monitoring Server attempts to delete unnecessary data taking up space in the medication dispenser using DELETE commands. In the case of software error, the Monitoring Server recovers the error through the configuration or software management operation.

5. Implementation and Verification Results

5.1. Implementation Results. Figures 6 and 7 depict the printed circuit board (PCB) and a prototype of the smart medication dispenser, respectively.

As shown in Figures 7(a) and 7(b), the number of MDTs in the smart medication dispenser can be increased. Figure 7(a) shows the front view of the smart medication dispenser with one MDT (i.e., for one patient), whereas Figure 7(b) shows the smart medication dispenser with two MDTs (i.e., for two patients). The smart medication dispenser can be connected to a medication-monitoring server via an Ethernet port. It can also be connected to the server wirelessly by attaching a wireless modem or a Bluetooth dongle to the USB port. In addition, the dispenser also supports RS232 serial communication with a local PC.

Figure 8 depicts the graphic user interface (GUI) of the smart medication dispenser. It is displayed through the touch-sensitive LCD.

Figure 8(a) shows the main view, which displays the current date and time, user name, type and time of upcoming medication, and notices. It also contains the “System Config” menu for changing the medication schedules and system settings. Figure 8(b) depicts the medication schedule

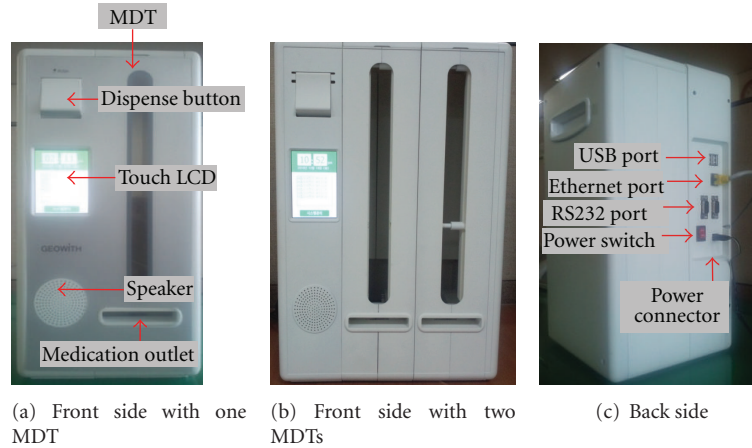


FIGURE 7: Prototype of smart medication dispenser.

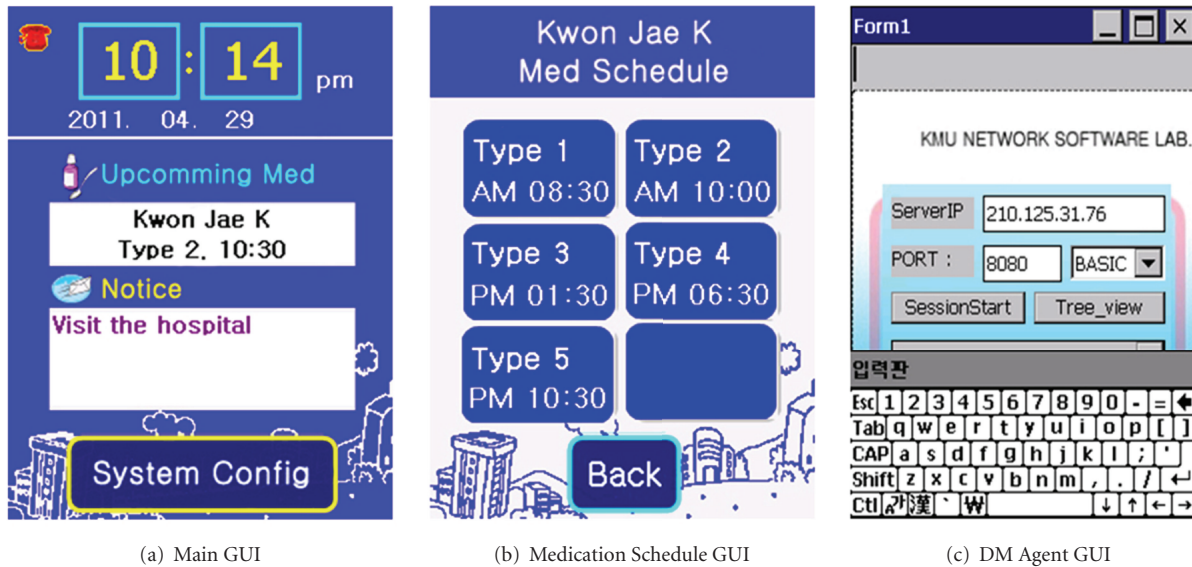


FIGURE 8: Smart medication dispenser GUI.

GUI. The user can check the configured schedules and change them through the touch-sensitive LCD. Figure 8(c) represents the DM Agent GUI. The user can change the settings related to the Monitoring Server such as its IP address, port number, and authentication method through this GUI. The smart medication dispenser supports two types of authentication: BASIC and MD5. The GUI also contains the “Session Start” menu, which allows the user to initiate a management session manually, and the “Tree.view” menu to check the DM Tree and MOs. These programs are implemented in C# and the sizes of the firmware, DM Agent, and DM Daemon were 3.62 MB, 152 KB, and 40 KB, respectively.

5.2. Verification Results. To verify the typical operation of the proposed medication dispenser, the Funambol server [29] based on the OMA DM protocol is used as a central DM server. The server provides the Web-based interface

for generating management commands. First, we verify the system settings management operation as shown in Figure 9.

The figures on the left show the screen of the medication dispenser, and those on the right show the screen of the Funambol DM server. In Figure 9(a), the medication schedule is configured as Type 0: 08:30 AM, Type 1: 10:00 AM, Type 2: 01:30 PM, Type 3: 06:30 PM, and Type 4: 10:30 PM. To modify the schedule, the server generates the following management command: <REPLACE,./Medication/TAKE/CH1,0070010900212003190042100>. This command replaces the values of the node./Medication/TAKE/CH1 (i.e., the medication schedule) with the following new values: Type 0: 07:00 AM, Type 1: 09:00 AM, Type 2: 12:00 PM, Type 3: 07:00 PM, and Type 4: 09:00 PM. In Figure 9(b), the medication schedule configured on the medication dispenser is updated suitably. In addition, the agent programs return status code 200 (OK) according to the server’s UI.

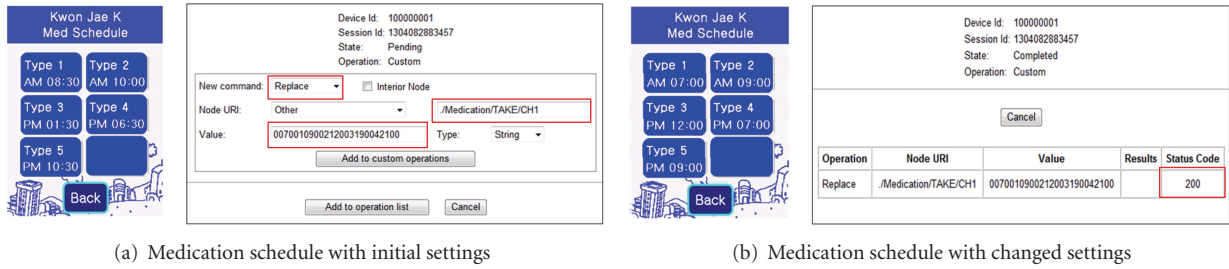


FIGURE 9: Example of system settings management operations.

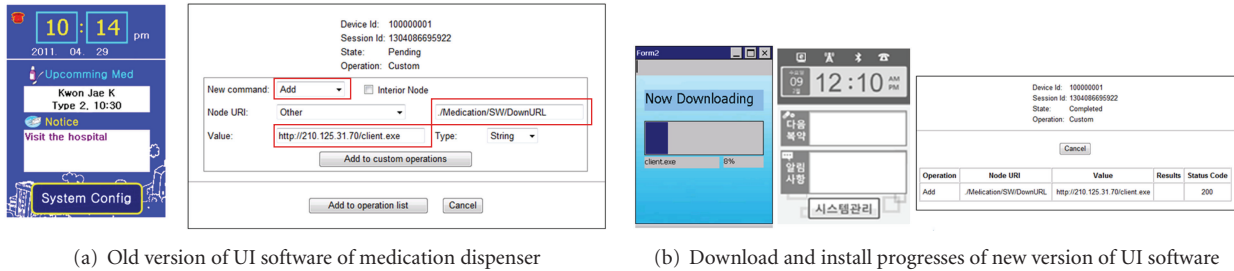


FIGURE 10: Example of embedded programs management operations

Next, we verify the embedded programs management operations as shown in Figure 10. As shown in Figure 10(a), the old version of the UI software is installed in the medication dispenser. Before updating the software, the server generates the following management command: <REPLACE,./Medication/SW/DownURL, http://210.125.31.70/client.exe>. This command initiates the download of a new version of the UI software from the URL, and installs the downloaded software. Figure 10(b) shows that the agent program downloads the UI software, and installs the new version of the UI software according to the server’s command. Based on the figure, the UI software installed on the medication dispenser is updated suitably. In addition, the agent programs return status code 200 (OK) according to the server’s UI.

From these figures, we can verify that the medication dispenser performs the server’s management operations, and the dispenser conforms to the OMA DM protocol.

6. Conclusion and Future Work

In this paper, we have proposed the smart medication dispenser to overcome the problems of existing medication dispensers, such as their nonexpandability, inconvenience, low reliability, and communication inefficiency. The proposed dispenser has three advantages over existing medication dispensers. (1) To achieve a high degree of scalability, the medication-dispensing trays can be attached in succession, and therefore, a single dispenser can support multiple users. (2) To achieve a high degree of remote manageability and (3) to reduce management costs and efforts, remote management methods are designed and implemented. These methods facilitate updating of the medication schedule configured

in the smart dispenser. Further, system settings, embedded programs, and operational errors can be remotely managed by medical staff and system administrators. Results of implementation and verification showed that the proposed dispenser operates normally and performs the management operations from the medication monitoring server suitably.

The smart medication dispenser can be used to improve medication adherence. It prevents overdosing, misdosing, and underdosing. However, it cannot prevent voluntary nonadherence, such as pretending to take medication or spitting it out afterwards. For future work, we plan to develop additional functions that detect a patient’s motions using a camera sensor to verify actual compliance. We also plan to extend our method to apply the smart medication dispenser to other personal health devices such as activity monitors.

Acknowledgment

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF), funded by the Ministry of Education, Science and Technology (no. 2012-013549)

References

- [1] World Health Organization, “Adherence to long term therapies: evidence for action,” <http://www.who.int/>.
- [2] R. L. Ownby, “Medication adherence and cognition: medical, personal and economic factors influence level of adherence in older adults,” *Geriatrics*, vol. 61, no. 2, pp. 30–35, 2006.
- [3] Pan American Health Organization, “Poor adherence to long-term treatment of chronic diseases is a worldwide problem,” *Revista Panamericana de Salud Pública*, vol. 14, no. 3, pp. 218–221, 2003.

- [4] e-pill, <http://www.epill.com/medtime.html>.
- [5] P. O'Donnell, S. Crowell, G. Huffman, and N. Jones, "Automated medication dispenser system," Technical Report of Mississippi State University, 2002.
- [6] W. W. Changa, T. J. Sungb, H. W. Huang et al., "A smart medication system using wireless sensor network technologies," *Sensors and Actuators A*, vol. 172, no. 1, pp. 315–321, 2011.
- [7] I. Sachpazidis and G. Sakas, "Medication intake assessment," in *Proceedings of the 1st International Conference on Pervasive Technologies Related to Assistive Environments (PETRA '08)*, July 2008.
- [8] V. F. S. Fook, A. A. P. Wai, M. Jayachandran, B. Jit, P. H. Lee, and P. Y. L. Kiat, "Smart mote-based wireless medication management system for persons with dementia," *Assistive Robotics and Mechatronics*, vol. 9, no. 1, pp. 27–35, 2008.
- [9] J. Dunbar-Jacob and M. K. Mortimer-Stephens, "Treatment adherence in chronic disease," *Journal of Clinical Epidemiology*, vol. 54, no. 12, supplement 1, pp. S57–S60, 2001.
- [10] P. A. Frick, L. Lavreys, K. Mandaliya, and J. K. Kreiss, "Impact of an alarm device on medication compliance in women in Mombasa, Kenya," *International Journal of STD and AIDS*, vol. 12, no. 5, pp. 329–333, 2001.
- [11] S. F. Laster, J. L. Martin, and J. B. Fleming, "The effect of a medication alarm device on patient compliance with topical pilocarpine," *Journal of the American Optometric Association*, vol. 67, no. 11, pp. 654–658, 1996.
- [12] J. A. Cramer and R. H. Mattson, "Monitoring compliance with antiepileptic drug therapy," in *Patient Compliance in Medical Practice and Clinical Trials*, pp. 123–137, 1991.
- [13] Open Mobile Alliance (OMA), <http://www.openmobilealliance.org/>.
- [14] J. M. Silva, A. Mouttham, and A. El Saddik, "UbiMeds: a mobile application to improve accessibility and support medication adherence," in *Proceedings of the 1st ACM SIGMM International Workshop on Media Studies and Implementations that Help Improving Access to Disabled Users (MSIADU '09)*, pp. 71–78, October 2009.
- [15] W. Mei-Ying, P. H. Tsai, J. W. S. Liu, and K. Z. John, "Wedjat: a mobile phone based medicine in-take reminder and monitor," in *Proceedings of the 9th IEEE International Conference on Bioinformatics and BioEngineering (BIBE '09)*, pp. 423–430, June 2009.
- [16] J. M. Slagle, J. S. Gordon, C. E. Harris et al., "MyMediHealth—designing a next generation system for child-centered medication management," *Journal of Biomedical Informatics*, vol. 43, no. 5, pp. S27–S31, 2010.
- [17] E. Becker, V. Metsis, R. Arora, J. Vinjumur, Y. Xu, and F. Makedon, "Smartdrawer: RFID-based smart medicine drawer for assistive environments," in *Proceedings of the 2nd International Conference on Pervasive Technologies Related to Assistive Environments (PETRA '09)*, pp. 1–8, June 2009.
- [18] S. Ammouri and G. A. Bilodeau, "Face and hands detection and tracking applied to the monitoring of medication intake," in *Proceedings of the 5th Canadian Conference on Computer and Robot Vision (CRV '08)*, pp. 147–154, May 2008.
- [19] D. Batz, M. Batz, N. D. V. Lobo, and M. Shah, "A computer vision system for monitoring medication intake," in *Proceedings of the Canadian Conference on Computer and Robot Vision (CRV '05)*, pp. 362–369, May 2005.
- [20] J. Case, M. Fedor, M. Schoffstall, and J. Davin, "A Simple Network Management Protocol (SNMP)," RFC 1157, IETF Network Working Group, 1990.
- [21] Distributed management task force, web-based enterprise management (WBEM), <http://www.dmtf.org/standards/wbem>.
- [22] P. Jugeon, P. Keehyun, J. Daejin, J. Myungsook, and W. Jongjung, "Design of DM agent based on the WIPI," *Journal of Mobile Technologies, Knowledge and Society*, vol. 4, pp. 61–67, 2007.
- [23] P. Jugeon, P. Keehyun, J. Daejin, and J. Myungsook, "Design and implementation of wireless device management agent based on OMA DM," *Korean Institute of Information Scientists and Engineers*, vol. 14, pp. 363–368, 2008.
- [24] L. Jieun, S. Sunghak, and J. Byungduck, "WiBro device management system based on OMA DM protocol," *KNOM Review*, vol. 10, no. 2, pp. 1–11, 2007.
- [25] K. Joonmyung, J. Hongtaek, C. Mijung, W. H. James, and K. Jungu, "OMA DM-based remote software fault management for mobile devices," *International Journal of Network Management*, vol. 19, no. 6, pp. 491–511, 2009.
- [26] K. Joonmyung, J. Hongtaek, C. Mijung, and W. H. James, "OMA DM-based remote software debugging of mobile devices," *Managing Next Generation Networks and Services*, vol. 4773, pp. 51–61, 2007.
- [27] C. Mijung, W. H. James, and J. Hongtaek, "XML-based network management for IP networks," *ETRI Journal*, vol. 25, no. 6, pp. 445–463, 2003.
- [28] R. Hyunki, C. Sungrae, P. Shiquan, and K. Sungho, "The design of remote vehicle management system based on OMA DM protocol and AUTOSAR S/W architecture," in *Proceedings of the 7th International Conference on Advanced Language Processing and Web Information Technology (ALPIT '08)*, pp. 393–397, July 2008.
- [29] Fumanbol DM Server, <http://funambol.com/>.